A Multiple Objective Evolutionary Algorithm for Multiple Sequence Alignment

Pasut Seeluangsawat
Department of Computer Engineering,
Chulalongkorn University, Bangkok, Thailand.
g46psl@cp.eng.chula.ac.th

Prabhas Chongstitvatana
Department of Computer Engineering,
Chulalongkorn University, Bangkok, Thailand
prabhas@chula.ac.th

ABSTRACT
The problem of multiple sequence alignment is important for bioinformatics. This problem is widely studied and a popular tool to solve this problem is Clustal X. This work introduces a multiple objective evolutionary algorithm to improve solutions obtained from Clustal X. The proposed method is tested with the dataset from BAliBASE database.

Categories and Subject Descriptors
J.3 [**LIFE AND MEDICAL SCIENCES**]: – *Biology and genetics, Medical information systems.*

General Terms: Algorithms, Performance.

Keywords: Multiple objective evolutionary algorithm, Multiple sequence alignment, Clustal X, BAliBASE.

1. INTRODUCTION
Multiple sequence alignment (MSA) of amino acid sequences is a fundamental part of bioinformatics and dynamic programming is a popular method to solve MSA problems. Many previous works for example [1] used sum-of-pair to scoring and improving the solution. This work introduces the use of multiple sets of penalty function in an evolutionary algorithm to optimize the alignment.

2. The MOMSA Algorithm
MOMSA is described as follow: let P be a population, A be nondominated set or archive, T be maximum number of generations.

I. Generate an initial population P_0 from Clustal X, set empty archive A_0 and $t = 0$.
II. Calculate objective value and rank value of individuals in P_t and A_t.
III. Select individual in P_t and A_t to A_{t+1}.
IV. If $t \geq T$ then stop.
V. Use binary tournament selection on P_t and A_t.
VI. Use recombination and mutation operators to create P_{t+1}, set t=t+1 and go to step II.

GECCO'05, June 25-29, 2005, Washington, DC, USA.
ACM 1-59593-010-8/05/0006.

The input in MOMSA is derived from the output of program Clustal X [2]. The new archive is derived from nondominated individuals (rank = 0) and they must not be the same individuals. The binary tournament selection is used to select individuals. Recombination and mutation operators are used to create new individuals. Finally, the program repeats step II and checks for the condition to stop or to terminate.

2.1 Initial Solutions
The output from Clustal X is used as an initial solution in the MOMSA population. The sequence is extended 10% longer than the output from Clustal X.

2.2 Objective Function
To evaluate the score of a candidate alignment, the reward function and the penalty function are used.

$$Objective\ \ function = SPscore - GapPenalty \qquad (1)$$

SPscore is sum-of-pair score. There are several cost matrices such as PAM, BLOSUM to use in calculation the cost. Each cost matrix indicates the probability of the similarity between residues. Blosum45 is used in this work.

$$SPscore = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} BLOSUM45(l_i, l_j) \qquad (2)$$

The gap penalty score specifies the penalty score.

$$GapPenalty = GOP + GEP \times GAPS \qquad (3)$$

where GOP was gap opening penalty, GEP was the gap extension penalty and GAPS was the number of consecutive gaps under consideration.
MOMSA solves this problem by using two objective functions. The first objective function was assigned value by GOP>GEP and the second objective function was assigned value by GOP<GEP.

2.3 Pareto ranking
Each individual i in the population P_t and archive A_t is assigned the rank value.

$$Rank(i) = \left| \{ j \mid j \in P_t + A_t \wedge j \succ i \} \right| \qquad (4)$$

Where $\left| \cdot \right|$ denotes the cardinality of a set, $+$ stands for multiset union and $\succ$ is the Pareto dominance relation.
The rank zero means that the individual is nondominated solution.

2.4 Environment Selection

The new archive is derived from two conditions. First all nondominated individual in P_t and A_t (rank = 0) are copied to the new archive.

$$A_{t+1} = \{i \mid i \in P_t + A_t \wedge Rank(i) = 0\} \quad (5)$$

Second, each individual in the new archive must not be the same.

2.5 Operators

In MOMSA, various variational operators are used to improve solutions. One recombination operator and three mutation operators are used in this work. The Recombination operator is TwopointCrossover and mutations are MoveColumn, ShiftSide and RandomLocalShuffle.

The TwopointCrossover randomly chooses column region that the positions of residue are not the same in two old individuals and exchanges the chosen column region. The MoveColumn operator randomly selects a column that residues in this column are near gaps and chooses a random direction to move the column left or right. Residues in the selected column follow the direction of the last position in the gap region. The ShiftSide operator selects a random gap region from a randomly chosen row and moves residue near the first or last gap in the chosen gap region to the other side. If the gap region has residue in the left and right hand side then randomly chooses a side. The RandomLocalShuffle operator picks a random residue from a randomly chosen row and checks a gap region in its neighbor. If there is a gap region, this operator exchanges the selected residue with a randomly chosen position in the gap region. Finally, the program will always pack gap columns to the end of last column.

3. Experiments

3.1 BAliBASE

This work used nine data sets (1taq, 1aad, 1pii, 1pfc, 1hfh, 451c, kinase, 1aboA, 1tvxA) from the first reference in BAliBASE [3] database in the experiments. In Table 1, NSEQ are number of sequences, LSEQ are length of sequences and SEQID are percent residue identity.

3.2 Evaluation measure

The result from the MOMSA is compared with the BAliBASE reference alignments using the BAliBASE evaluation measure. The sum-of-pair score (SPS) is calculated such that the score increases with the number of sequence correctly aligned. The column score (CS) counts the number of correct alignment of all sequences.

3.3 Experimental Setup

In this study we used the default parameter setting provided with Clustal X to generate the candidate input for MOMSA. The population size is 50 and the total number of generation is 200. BLOSUM45, substitution matrix, is used. The first objective function has GOP=10 and GEP=1. The second objective function has GOP=8 and GEP=12. The probability of recombination operator was 0.25 and mutation operators were 0.75. Three mutations were randomly chosen with equal probability.

4. Result

Table1 summarizes the performance of MOMSA on the test set. The SPS and CS columns show the BAliBASE scores. The best column shows the score derived from overall best alignments. The mean and SD columns show the mean score obtained from 30 runs of program and standard deviations.

The MOMSA improved the SPS score of all data sets. In CS score, mean score was improved in seven data sets over 30 runs. The remaining two data are similar or equal to Clustal X. The standard deviations were low. This indicates the reliability of MOMSA in obtaining solutions.

5. REFERENCES

[1] R. Thomsen, G. B. Fogel, and T. Krink. Improvement of Clustal-Derived Sequence Alignments with Evolutionary Algorithms. Proceedings of the Fifth Congress on Evolutionary Computation, 2003.

[2] J. D. Thompson, T. J. Gibson, F. Plewniak, F. Jeanmougin, and D. G. Higgins. The ClustalX windows interface: flexible strategies for multiple sequence alignment aided by quality analysis tools. Nucleic Acids Research, vol. 24, pp. 4876–4882, 1997.

[3] F. Plewniak, J.D. Thompson, and O. Poch. BAliBASE: a benchmark alignment database for the evaluation of multiple alignment programs. Bioinformatics, vol. 15, pp. 87–88, 1999

Table 1. Comparison between Clustal X and MOMSA in terms of SPS and CS.

Dataset	Clustal X		MOMSA					
	SPS	CS	SPS			CS		
	*	*	Best	Mean	SD	Best	Mean	SD
1taq	0.874	0.810	0.878	0.876	0.001	0.819	0.817	0.011
1aad	0.818	0.696	0.833	0.833	0.00	0.714	0.714	0.000
1pii	0.787	0.618	0.789	0.789	0.002	0.622	0.621	0.004
1pfc	0.774	0.600	0.797	0.784	0.007	0.620	0.603	0.006
1hfh	0.820	0.624	0.836	0.829	0.004	0.679	0.661	0.01
451c	0.555	0.338	0.568	0.561	0.003	0.354	0.354	0.000
kinase	0.655	0.485	0.665	0.661	0.003	0.494	0.488	0.003
1aboA	0.693	0.556	0.713	0.706	0.006	0.556	0.534	0.005
1tvxA	0.223	0.000	0.227	0.225	0.002	0.000	0.000	0.000